DOCUMENT RESUME

ED 249 676 EC 170 472

AUTHOR Lyons, Dave
TITLE A Computer Aided Implementation of Precision Teaching.
PUB DATE 84
NOTE 27p.; In: Communication Problems of the Mentally Handicapped. Occasional Paper No. 28; see EC 170 469.
PUB TYPE Speeches/Conference Papers (150) -- Reports - Descriptive (141)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Software; *Precision Teaching; *Program Development; *Severe Mental Retardation

ABSTRACT

A computer implementation of precision teaching for severely mentally handicapped students is described. Computer technology reduces staff time during a typical precision teaching routine: design of suitable teaching programs, a series of timed short drills using these programs, conversion of results into numeric quantities, and design of improved programs from the numeric quantities obtained. Computer applications in recordkeeping are noted. Specific information on hardware address such aspects as projector, switch and clock controller, and the voice synthesizer controller. Similarly, information is detailed on the file system and computer software, including the synthesizer unit, the statistics package, and the library and date units. It is concluded that the system appears to be a success, and that it is both easy to use and reliable. (CL)

**********
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
**********

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION
CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

P. T.
Culhane

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ED249676

# A COMPUTER AIDED IMPLEMENTATION OF PRECISION TEACHING

DAVE LYONS (Department of Computer Science, University of Essex)

Abstract

The Precision Teaching method, which is a proven teaching mechanism for the severely mentally handicapped, is extremely staff-intensive in use. This paper describes a computer implementation of the method which is not staff-intensive, and provides a record keeping and archiving facility.

The system has three human interfaces; the handicapped subject, the care staff, and the psychology staff. The handicapped subject is drilled by one of a set of generic games. The subject sees projected pictures, hears (synthesized) utterances, and responds by manipulating toys. All parameters for a given subject are individual, and are changeable by the care staff, who see a console which guides them through the system's use (including such functions as backup). Very little training in its use is needed. The pyschology staff see a statistical printout, from which they may require modifications of the parameters.

The current equipment consists of a voice synthesizer, a high speed random slide projector and a set of special buttons attached to a 64K Z80 with two disk drives, but the system will run on many microcomputers. The system is complete and is about to go into service.

The project has been funded by Project Horizon, a charity associated with Turner Village Hospital, Colchester.

EC170472

Acknowledgements

The author would like to thank Project Horizon for the funding, but more important, the encouragement and focus for this work. The Department of Computer Science at the University of Essex has provided technician effort and space for the development of the system, and has gifted its word processing software to the project.

Many teachers in Alberta, particularly M Thompson of Lethbridge, have given me access to their classrooms, and suffered the ignorance of a computer scientist. The Turner Village staff, headed by Dr R McKibben, have similarly suffered.

It is they, the psychiatrists, psychologists, educators and particularly the care staff that are providing the heart of this project.

Most of all, I must thank the residents of Turner Village Hospital, Colchester. It is they who will teach us how to design this or any other teaching method.

## 1 Introduction

Project Horizon is a charity associated with Turner Village Hospital, which is a hospital for the mentally handicapped in Colchester, Essex. The purpose of Project Horizon is to provide specialised facilities to 'raise the horizons' of the residents. These include specialised adventure playgrounds, protected farm environments and the like, which are very well used and considered to be of considerable benefit to the residents. The Specialist Committee of the charity is charged with applying modern technological solutions to some of the problems seen at the hospital. It is from this base that the present work has grown.

The psychology staff can measure with usable accuracy the current and potential capability of a resident under a set of criteria. These criteria include gross and fine motor control, oral and aural capabilities, social skills, symbol manipulation and attention span. These latter two are particularly important for the present work. It is usually the case that a currently low ability in one area severely limits raising ability in another. As an example, a person with a very low attention span is extremely difficult to teach, even though the capacity to learn may be there. If the attention span is increased, then the possibility of increasing performance in other areas is much greater.

In any further program, analysing the results of teaching so as to design further programs is difficult. One of the best teaching mechanisms to aid this is Precision Teaching (Lindl). Basically, Precision Teaching consists of the following stages: the design of putatively suitable teaching programs for a resident; a series of timed short drills using that program; the conversion of the results into numeric quantities; and the design of improved programs from the numbers. The mechanism will be dealt with in Section 2, but note that most of the staff time is spent in paperwork, manipulating results, and the redesign of the teaching programs. The latter would have to take place in any good teaching method, and, indeed, can be much more accurately and easily done with this one. Nevertheless, most of the staff time must be spent away from the resident doing paperwork. This is a major failing of the method.

The Project Horizon system sets out to overcome this failing. It is basically a teaching machine and record keeper. It drills the resident, collects results and timings, does the

maths and presents results. The drills consist of one of a set of 'games', in which the stimuli are projected images and synthesized speech, and the responses are their manipulation of 'toys' mounted on special switches. As the computer is doing the timing and tallying, that load is taken from the staff. Further, the system calculates the statistics, and produces a metric from each drill and set of drills.

It is very important that each resident be treated as an individual; the system must not restrict everyone to the same set of parameters. For example, they may not use the same slides, and even if they do, they may be associated with different concepts, and have different grades of difficulty. In the Project Horizon system ALL parameters can be varied for each individual. This demands good cosmetics, or the staff will not use it; 'it may be flexible, but I don't know how to flex it!'. In fact, a major design criterion was to present clean and friendly interfaces to the three user communities; in this case, the patients, the care staff, and the psychology and educational staff. In particular, they need not know about concepts of the underlying system such as files. It is, in effect, a turnkey system. The design criteria are discussed in Section 3.

Because of fund limitations, it was necessary to restrict the pilot project to individuals with some gross motor control, sight, and hearing. The stimuli are pictures and sounds, and the response is a manipulated toy. The staff can answer for the patient, and then the system is acting as a sophisticated recording device.

Included in the criteria is the need to provide a history of the progress of an individual. The system provides a set of comment facilities, so it can as well be used as a medical notebook. The need for history implies an archive facility, and an easy retrieval mechanism. In an attempt to pervert Murphy's law, a backup facility is also included, and is semi-automatic.

The hardware costs are quite low. The project funding is $5400. A Vector MZ Z80 based 64K byte micro-computer with two disk drives, a terminal, a printer, a voice synthesizer, and a random access slide projector, together with their controllers, make up the hardware. These are discussed briefly in Section 4. The special controllers are S100 bus devices. They are not complex, so a redesign to another interface would not be difficult.

The software is described in Section 5. It was written in UCSD Pascal (Pascl), and consists of a main program and twelve units, and some auxiliary support programs. It is important that the program structure be good, as modifications to the system over time are inevitable. Further, as funds become available to ease the restrictions on the abilities of the individuals served, new stimulus and response tools will have to be added without disturbing the system as a whole, or consuming much programming time. Some conclusions on this subject are made in Section 6.

At the time of writing, the system was in the process of being delivered. The training of the operating staff, that is, the care staff and the psychologists, has been completed, and took only a short time. Some successful trial runs with children at Turner Village Hospital were made. These were, of course, primarily to help the implementor; even so, the attention span of one child was increased considerably. If the equipment can help the patients to 'raise their horizons', then it will be a success.

## 2 Precision Teaching

The basic principles of Precision Teaching grow from early research into the behaviour of lower animals in the 1930s. In the 1960s, Lindsley and his students at the University of Kansas began in intensive effort to make the principles applicable to children with mental handicaps, in a form relevant and accessible to teachers (Lind2). He particularly stressed that 'retardation' was not a useful term in the lexicon of teachers of these children (Lind3). Since then, much work has been done, and the method is in use in many centres (Altal, Montl, Batel). There is no doubt that it can be extremely successful (Meffl, Bate2, Perrl).

Precision Teaching is the application of the principles of operant conditioning to educational problems (Pittl, Hirsl). Operant conditioning is concerned with the relationship between behaviour and the environment, and attempts to understand this relationship by gaining knowledge of the factors that modify behaviour. To be useful, some metric must be available, so that results can be compared. Operant conditioning refers to the process in which the frequency of occurrence of a particular behaviour is modified by the consequences of that behaviour.

The Precision Teaching cycle is as follows:

1 Select a behaviour that needs to be strengthened, weakened or modified. In selecting a behaviour, it is important that it be concrete. Terms such as 'attention span' are useful. Terms such as 'intelligence', 'dyslexia', 'emotional disturbance' are not

2 Choose a metric. A typical metric used to judge the efficacy of a program is response rate, ie, responses divided by session time

3 Construct a teaching program

4 In a fixed instructional environment, implement the teaching program. During a teaching session, the teacher would have to record tallies of 'correct' and 'incorrect' responses, and the times at



BEST COPY AVAILABLE

which they occurred

5 Analyze the data, and modify the program if necessary. The analysis of the metric involves graphing it for several sessions, and analyzing the resultant curve

An important feature of the method is the instructional environment. It must be constant over time, or it will interfere with the behaviour.

It can be seen that there is considerable work that must be done without the child present. The design and construction of teaching programs is an example, but this would (or should) be the case with any teaching method. The author has been privileged to watch sessions in schools in Alberta. The timing and tallying is time consuming, and can influence the design of a program. The analysis is time consuming, and takes the teacher away from the child. It is one of the aims of the work described in this paper to reduce this 'non-contact' time to the absolute minimum.

## 3 System Design Criteria

In setting the design criteria, it was clear that three quite separate 'audiences' were involved: the operating care staff and the psychology staff. Of course the residents must be provided with an interesting environment, and the computer equipment itself must not be part of it. It must be safe. It must provide a good interpretation of the Precision Teaching method. All of the system parameters must be variable for each individual.

The care staff and the psychology staff presented more of a challenge as users. One must assume that they are not familair with computers. The system must be simple to run, and the training period brief. A short training period, and a flexible system are antithetical requirements.

Funding limitations must of course be taken into account. In this Section the design criteria are described, and in Section 6 there is a short discussion as to how well they were met.

### Stimuli and Responses

The range of human stimuli is immense. The problem here was one of restriction for a pilot project. Visual stimulus was chosen, as it is fairly cheap. Some of the residents have extremely short attention spans, so changes in stimulus must be rapid. Computer graphics were ruled out as being too abstract. A normal slide projector can provide good images of reality, and creating the images is a simple matter of taking a picture. The

images required are not sophisticated; they must be clear and simple. However, a normal slide projector is much too slow (up to forty seconds) in selecting a random slide. A random access projector, the Kodak RA2000 (Kodal) was chosen, which has an interslide time of about four seconds maximum.

In working with the residents, and in discussion with the psychologists, it became clear that visual stimulus was not enough, and that vocal stimulus was necessary. Again, there is the problem of instant access to an utterance. Tape would provide good quality sound, easy to record; however, selecting a random utterance would either be very slow, or require special, expensive equipment, probably custom designed. Digitized speech would be fast enough, but storage and cost become problems.

An advantage of synthesized speech is data compression, allowing instant access to a random utterance. The problem then becomes the implementation of a voice synthesizer with a reasonable quality. The Votrax SC01A (Votrl) was chosen. It is cheap, and easy to control. A cosmetic for creating utterances then has to be contructed, so the staff can create utterances easily. Given the synthesizer, one might consider text-to-speech techniques (Elovl). Currently, automatic intonation construction is an unsolved problem; as intonation carries much more of the information in an utterance for a resident. This technique is ruled out.

In responding to a stimulus, the resident must provide an input to the system. The input device must be instantly recognisable as an object in the residents normal environment. These objects will vary from resident to resident, as a resident's abilities grow, or the educational aims vary for her. Whatever mechanism is used, it must be absolutely safe, as this is the only part of the system the resident will touch. A pneumatic industrial switch was found that completely isolated the resident (in the electrical sense) from the equipment, and is quite cheap. Toys, pictures or other objects are attached with Velcro. The pressure required to activate the button can be varied within a reasonable range.

Precision Teaching Programs

The Precision Teaching Programs in this system are a set of generic 'games'. Examples:

> Class match games, in which the images must match the toy for a correct response. There will be various types of these with different parameters, and different numbers of classes and difficulties. The method of changing classes and difficulty must be a parameter. A one class game should be allowed, in which the subjects simply changes their environment when they perceive a stimulus.

Yes-no games, in which questions are asked (by an utterance) about the slide images. As well as the parameters for class match games, there will be some to control whether a 'yes' or a 'no' is the correct answer.

Some subjects (self-mutilating children as an example) require special utterances from time to time during a session to help to prevent behaviour which interferes with the training. In all games, the length of the game, the 'no response' timeout, whether a 'special' utterance is required, should be parameters. The definitions of the parameters for each generic game should be kept on auxiliary files with their default values.

In Precision Teaching, data from a single session is less interesting that data across sessions. Session and statistical data should occur in distinct sets.

The creation of new generic games should not require the skills of a computer scientist.

Record Requirements and Accessibility

One can not decide ab initio what information is transient, and what is to be kept. All information must be maintained, and if a change is made, the old information is held. This implies a mechanism for superseding information; that is, making it inaccessible for current use, but leaving it 'on file' for later access if required. Current information should be quickly accessible, and superseded information should be accessible by date. Each item, then, must be date and time stamped.

Information is kept in records. The following record types must be available:

Biographical information, such as name, nickname, age, and birthdate

Raw session data should be kept. Even if a compressed metric of a session with a subject is stored, one may wish to go back to the raw data

Statistical compressions of session data must be stored in such a way that they can be related to the raw data at need

The particular stimulus and response objects must be recorded in detail. Thus records to contain detailed slide, utterance and 'toy' information must be kept. To save space, textual descriptions of slides, utterances and toys, and their default characteristics, should be kept in auxiliary files, separately maintained

The parameters for the game played will vary in detail from game to game, from subject to subject, and over time with a given subject

To a programmer a string of characters is just that. The operators, however, will wish to distinguish among a class of comments. There will be subject-general comments, demographic descriptions, medical data and comments on sessions and progress. These must be separately accessible

Modification of Subject Information

Not all data should be modifiable by the operator. Raw session data and past statistical compressions should remain untouched. The rest of the data should be modifiable, and the cosmetic will vary with record type. The description of some items may change. For example, if a different game is played, different parameter descriptions will be necessary. These descriptions should be kept on auxiliary files, separately maintained. Similarly, descriptions of slides (on a tray by tray basis), toys and utterances should be on auxiliary files, with their default parameters. This information should be displayed during the modification procedure.

The cosmetic for the construction of utterances will be the most difficult. Operators will have to 'take apart' words. For example, the word 'hello' needs the phonemes 'H', 'EH', 'L', 'O', and 'W'. The system should give them as much aid as possible in describing what is to be done to create or modify an utterance. There must be a mechanism for filing and retrieving utterances, so an utterance 'bank' can be available.

System and Data Integrity

Data can be lost through operator error, or through machine malfunction. The system must cope with both, and be designed so as to make operator error as unlikely as possible. Data in questionable files must not be accessible with the normal access mechanisms until it is fixed or declared 'healthy'. If access by the staff to the underlying constructs, such as files and directories, is required, operator error is likely.

The system must be a 'turnkey' device. The operators must not need access to filers, editors, etc. Indeed, they should not be given access to them. They must view the system as a set of 'subjects'.

Nevertheless, problems may occur. Checks must be placed in the data. If a check fails, then the data must be marked unsafe. In particular, if the program fails on an open subject, that subject must be marked unsafe.

BEST COPY AVAILABLE

BEST COPY AVAILABLE

In a real system subjects are implemented as files. Large files become unwieldy. An archiving mechanism must be present, and it must be transparent to the operator when requesting historial information. In a real world, data may be lost. A mechanism for backing up a subject must be part of the system.

General Cosmetic Considerations

With any new equipment, training considerations have to be taken into account in the design. The training period should be as short as possible; once the operators know how to turn it on, the ideal would be a 'self-training' program. It should be difficult to make a mistake.

The system must ask its questions of the operators in a uniform way. In particular, it must be clear what is information, and what is a demand for operator response. In general, an operator reply will be either of the 'yes-no' type, or will be a number or string. The former can be input by a single character, while the latter must have a terminator. The system must distinguish between them in its prompts for information. In the case of a 'yes-no' answer, 'no' must always be safe; only a 'yes' answer should cause action. Similarly, a null answer to a number or string evoking question should not cause any action. There should be no 'last chance syndrome'; if a 'no' answer to a question is given in error, the question should come round again.

Of course, NO answer should abort the program; for example, an alphabetic reply when a number is requested should cause a reprompt for the question, or indicate in some way the input was incorrect, and demand a new reply.

Program Structuring and Maintenence

As this is a pilot project, the time available to construct quite a large program is limited. The program will also be undergoing modification as new needs, and errors and limitations of the design are discovered. It is imperative that a debugging harness (inaccessible to the operators) is installed.

More important, all the rules of good structured programming must be applied. Full use of any structuring tools such as units must be made, so that separate testing and compilation is possible as new requirements are discovered. Interaction between units should be kept to minimum.

Statistical Processing

The statistical requirements will vary through time as the operators and the psychology staff learn about the system. The statistical package, must be both part of the system, and

changeable as new requirements are discovered. The ideal would be a generalized package, tailorable by the psychology staff. It is the latter group who primarily will be using this information, and it must be presented to them in an accurate, but palatable way.

## 4 The Hardware

The Vector MZ with two floppy disk drives was chosen for the system. An Epson MX80F/T Type II printer is also supplied. There are two special interfaces. The projector controller also provides the input port for the response switches, and a rudimentary clock facility. A controller is also required to drive the Votrax chip, and to augment its facilities. In addition, a small 10 watt amplifier and speaker enclosure was constructed.

None of the controllers is complex. They are designed to interface in a simple way to an S100 bus. However, redesign to another standard would not be a difficult task.

Safety was a key criterion. All devices are electrically isolated from the computer, either by opto-isolators, or capacitively in the case of the amplifier-speaker.

## The Choice of Computer

The Vector MZ is a Z80 based machine, with an 18-slot S100 bus. It comes complete with two single-sided double-density hard sector Micropolis 5 1/4" disk drives and controller, a memory mapped 80 by 24 character terminal, keyboard and controller, and parallel and RS232 ports. The screen has coarse (160 by 72 pixel) graphics. It is robust in construction, and very reliable.

The CPU board is not ideal; its treatment of S100 interrupts is rudimentary. It is really not the ideal choice for a real-time project. However, it had one overriding advantage: the University of Essex uses the Vector as an initial teaching and secretarial machine. The implementer had no learning curve to climb. In particular, UCSD Pascal was implemented on the configuration.

The Epson printer is a quiet, neat device, capable of graphic output. It is NOT letter quality, but is more than good enough for the statistical output required. In fact, the Department of Computing Science has agreed that the system can be delivered with our own word processing software.

The Projector, Switch and Clock Controller

The RA2000 slide projector has two imputs. The slide is chosen by two BCD digits. If the digits do not match the current position, the projector retracts the current slide, and moves to the new position. The second input is an indication to the projector as to whether it is to drop the slide for projection, or just hold it ready. The projector returns one three-level signal: one level indicates the slide is projected, one that it is retracted, and one that the projector is moving to a new position.

The internal impedences are low, and care must be taken in choosing the drivers. The isolation is done in the control box close to the projector, which also contains facilities for controlling lamp brightness and focus.

The switch interface is a simple parallel port. No debouncing is done in the controller. The isolation is achieved by the switch itself, which is a Radio Spares pneumatic industrial control switch (Radil). The isolation is given by three meter air tubes, which are covered with a tough non-toxic plactic coil to prevent injury to subject or equipment. They are very robust.

The clock is a simple sixteen bit register timing to 1/10 of a second, driven from the 2MHz S100 clock. This provides more than enough precision for the project. A non-volatile calendar board would have been preferable.

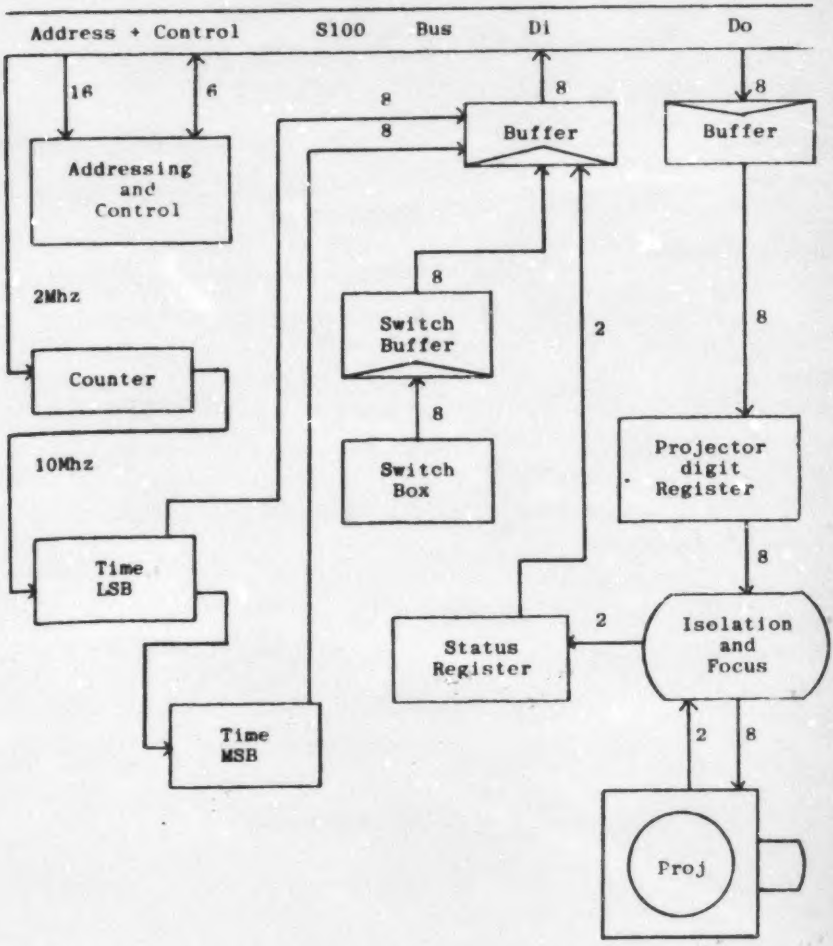
The Voice Synthesizer Controller

The Votrax SC01A synthesizer chip (Votr) is capable of creating sixty four phonemes. In fact, many of them are identical waveforms with different durations. It is clear (Elovl) that intonation contains much of the meaning of an utterance. In fact for some subjects, intonation contains all of the meaning. The chip can produce phonemes in one of four voices. More and finer intonation control is necessary, as the differences in the voice pitches is large.

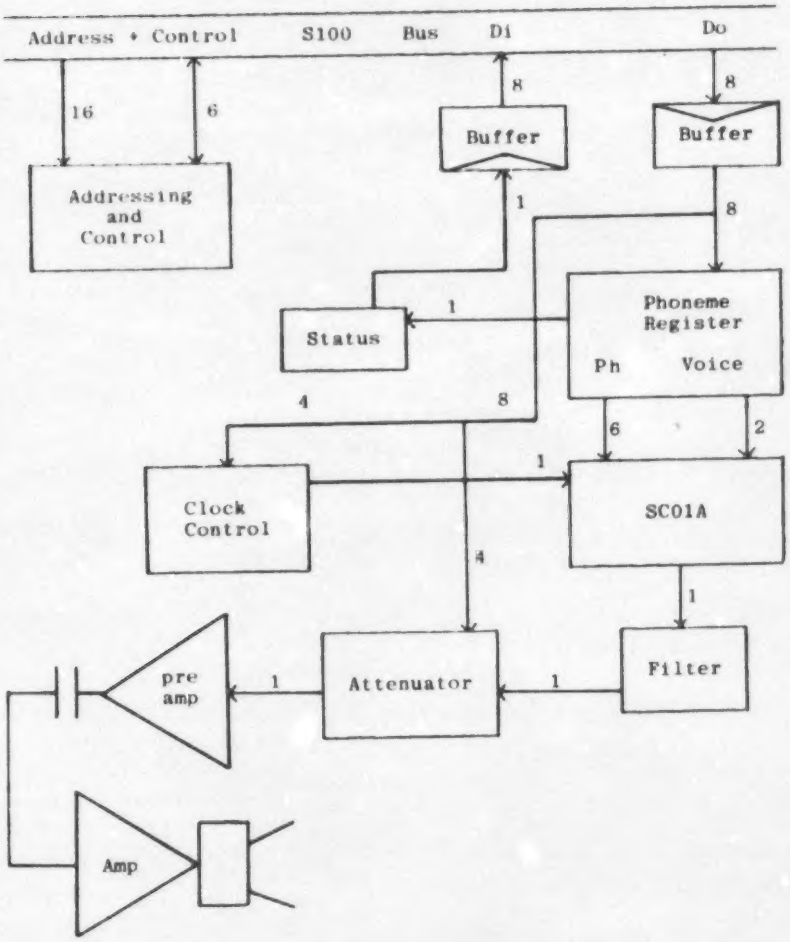
Fine frequency control can be obtained by varying the chip clock around its nominal frequency. This was done with an analogue resistor ladder controlled by a four bit register. The middle ten representations are useful. Volume control was implemented with another resistor ladder on the chip output, also controlled by a four bit register. Both controls are roughly logarithmic.

The controller takes two inputs from the S100 bus. The first contains four volume and four pitch bits. The second contains six phoneme bits and two voice bits, and activates the chip. There is one output, namely a completed indication from the chip.

Projector, Switch and Clock Controller Block Diagram



Votrax Controller



The chip output is buffered through a low pass filter and preamp, producing about three volts peak to peak maximum. The output is capacitively coupled to the amplifier.

## 5 The File System and Software

It was necessary to implement file structures which were flexible, relatively quick to access, and yet not overcomplex. There is one basic record, with variants for the various types required. Records of each type have to be accessed sequentially as if it were in a file of its own. The notions of 'current' and 'superseded' had to be implemented. It was most important that all data be kept, no matter how trivial the change. To support this, an archive system was implemented. A backup facility is also provided. There is auxiliary information kept and maintained separately; an example is slide information.

Like most programs of any size, most of the code is cosmetic in nature: that is, it supplies user friendly interactions for the operators. The heart of the system, running a session with a subject, is one of the smallest parts.

The system maintains one file for each subject, with archive files as the data grows, and backup files as time goes on. A name file contains short descriptions of subjects, and their main file names. Auxiliary files contain slide tray descriptions, toy descriptions, and game descriptions. Special files where single utterances may be stored are also maintained.

The system as supplied contains a main program, which maintains the subject files and name file, runs sessions, provides the parameter change mechanisms and print facilities, and does the statistics. An auxiliary program maintains the game, toy and slide files. The main program can maintain the utterance bank, but an additional program is supplied to do this 'offline'. A recovery program is available to correct files left in an unsafe state by bugs or machine failure.

UCSD Pascal is the implementation language. It has a separate compilation and overlay feature, UNITS, which allow large programs to be built. The rest of this Section briefly describes this facility, and the units constructed for the program. The auxiliary programs are also described.

### The Choice of Language

Pascal (Jensl) would not be the first choice of most implementers writing a real-time system. In fact, UCSD Pascal (Pascl) contains multiprocess facilities, including Djykstra's P and V functions (Dijkl). Interrupt routines, however, are difficult to install, and the interprocess communication is unsafe. If a deadly embrace occurs, a reboot of the entire

system is required.

The Unit feature, however, is very useful. Programs can be divided into separate compilation units. Each has an Interface Section, in which the function and variable declarations are made, as they are to be known to other units and programs using them. They have an Implementation Section, in which the functions are defined. A change in the Implementation Section requires a recompilation of that unit only. ADA (Adal) has a similar, if more powerful structure in Packages.

The present system is a pilot project. UCSD Pascal was primarily chosen because it did not require the implementor to install a language, and climb the learning curve of THAT language on THAT system.

The Turnkey Environment and the Chainer Unit

If the operator is given normal access to system facilities such as editors and filers, finger trouble is inevitable. The system must appear as a turnkey system. They must know, of course, how to load disks, and bootstrap. The first program run after the bootstrap is automatically SYSTEM.STARTUP. The startup program provided asks the current date, and then invokes the Chainer unit.

This unit is invoked at the termination of all the system programs. It displays a menu of descriptions of all of the programs available to the operator, and asks her to select one. The unit then chains to the requested program. In this way, the Pascal system itself is never seen by the operator. The currently available programs are:

the main program

the auxiliary file handler

an auxiliary voice synthesizer program

a file recovery program

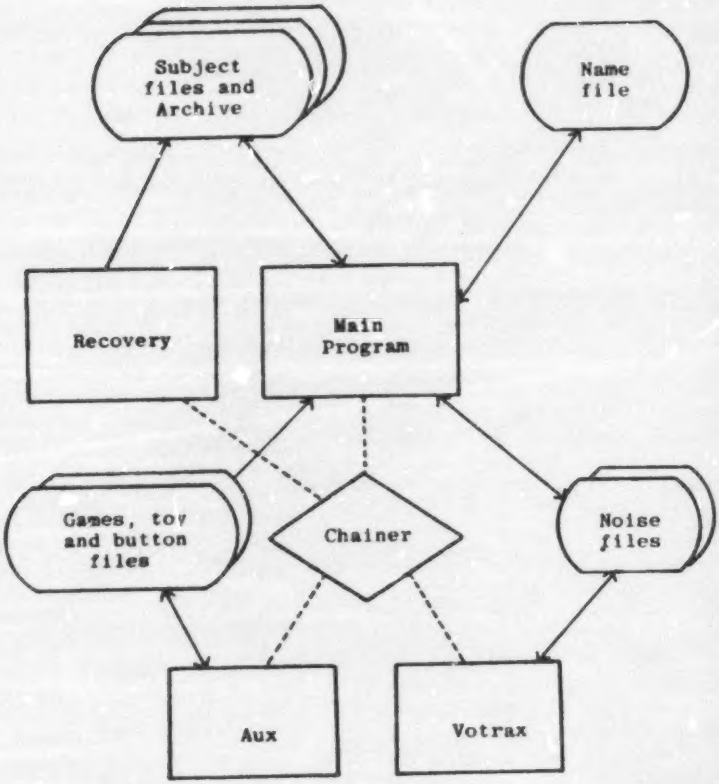
a 'toy' speech to text program

Of course, some mechanism must be available so that the maintainer can get into the system. Currently that choice is displayed on the menu as well, with a warning. As the chainer unit branches to all operator programs, we use this opportunity to show the relationships of the programs to the files, and to the chainer unit.

The Main Program, and Subject Selection

The first function of the main program is to ask the

Relationships of Programs to Files



operator to turn on the speaker and projector. It then displays a menu of currently known subjects. This information is held in the file NAMES.TEXT, which consists of pairs of lines; the first is a description of the subject, and the second is a disk and filename. The operator chooses a subject.

If the subject file is present on the mounted disks, the biographical information is displayed, and the program carries on to its next Section. If not, the reason is displayed. The operator is given a chance to try again or to name a file. If she chooses the latter, and no file exists, then she is asked if a new one is to be made, of what size, and whether the name and a description is to be added to NAMES.TEXT. There is no delete facility on the names file.

The program checks the date of the last backup operation. If it is past a week, the operator is asked whether the backup procedure is to be initiated. If there is little space left on the file, the operator is asked to initiate an archive procedure.

Once a subject file has been opened, the operator is asked cyclically which of the program facilities are required. These are:

information modification

information display and printing

running a system

statistical output

choosing a new subject

leaving the system through the chainer unit

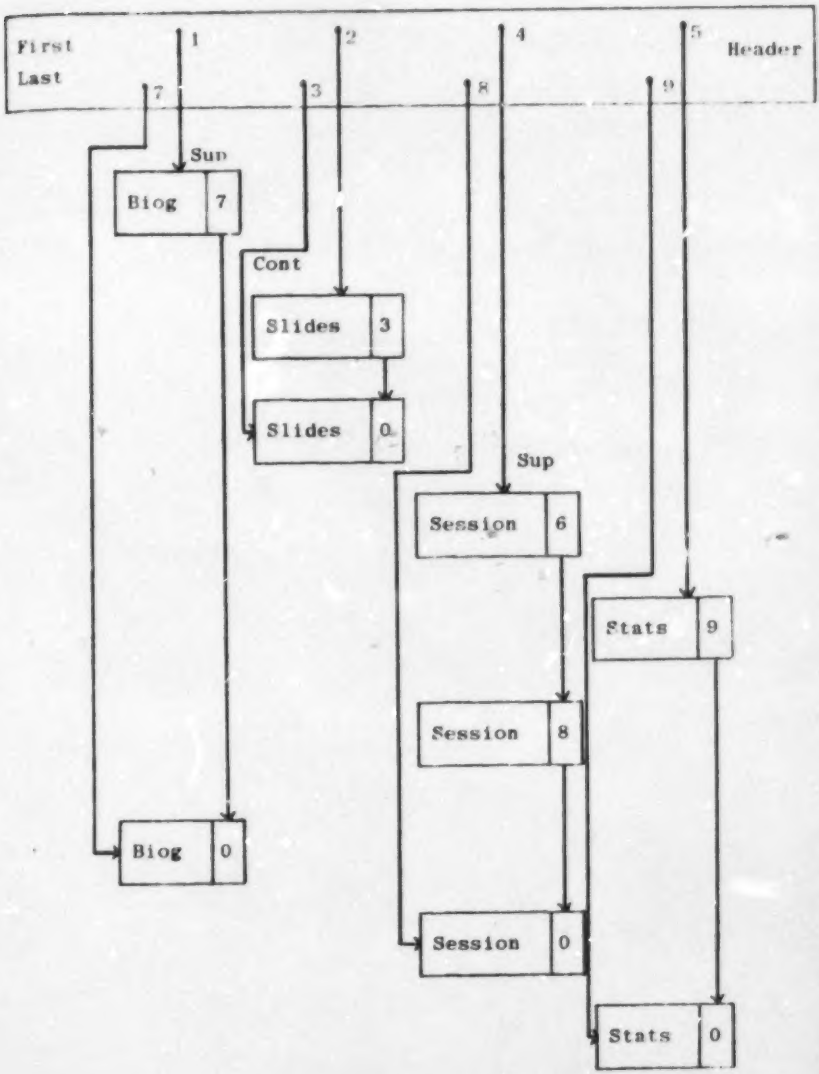
Subject File Definition and Manipulation

The subject file structure, and the routines for manipulating it are the province of the SUBFILEDEF unit. Its interface defines the record variants.

Each record has a common section containing a time and date stamp, the record address of the next record of this type, a superseded flag and a continuation flag. For some record types, it may be necessary for a logical record to be held in more than one physical record; comments are a good example of where this may be necessary. The rest of the record is a variant depending upon a record type flag.

Biographical records contain name, familiar name, date of birth, sex, and some psychological data about the subject. Comment records are basically character strings, but as the staff distinguish between comments (for instance, between comments

Example of Subfile Record Structure



about a session, and demographic comments), a subtype is stored.

Slides and toys have their own record types, in which the characteristics of the slide for this subject are stored. Both these types reference an auxiliary file, in which textual descriptions, and default values for the characteristics are kept.

There is a set of generic games, one of which a subject may be playing. The subject parameters will vary, and are kept in a suitable record. This record references an auxiliary description file. The utterances will also be tailored to the subject, and have a record type. These records each contain a phoneme string, and a textual description. However, the purpose of the utterance is held in the auxiliary file of the current game. An utterance whose purpose is affirmation may be 'good' for one subject, and 'terrific' for another.

The session record records the time of every event in a session: which slide, its class, and the response. For 'yes-no' games, the question sense is also recorded. This is the raw data for the statistical package, which creates statistical records for each session set.

The zero'th record of each file is a header. It contains backup and archive information, record counts, and a note of the earliest dated record of each type in the file to facilitate archive retrieval. It also contains the record numbers of the first and last record of each type in the file. As all records of a type are linked, all the records of one type appear as a sequential set, and can be so accessed.

The SUBFILEDEF unit contains the procedures that manipulate the file structure described above. The open routine does a special check for validity, and marks the file open. The close routine marks it closed. This ensures that, if the program crashes while a subject file is open, it becomes inaccessible to the program. The access routines include procedures to get the next record, check for end of 'file' for that type, 'rewind' to the first of a type, and write and rewrite record routines.

Subject Record Modification, Display and Printing

There are three entries into the SUBMP unit; record display and printing, record modification, and comment record creation. All three are used by the main program, but the latter is used where ever comments need to created.

The print and display entry asks the user to select an output medium, which can be printer, the console, or a named file. She can select to print current records only, records since some date (which may involve an automatic backtrack through the archives) or the whole subject file. IF the whole file is not to be printed (which is the usual case; this facility is used

for debugging only) then a menu of all record types is displayed, and a selection made. If comment records are selected, a subsidiary menu allows selection of one comment type, or all of them.

The record modification entry asks the user what current record type is to be changed (or created, for a new subject). The header, session, comment and statistical records may not be modified. Each modifiable type has a cosmetic to suit its data. Those that relate to auxiliary files have the auxiliary description data displayed. The records are NOT overwritten. The old record is superseded, and a new record supplied.

The comment creation entry is a simple device which allows a long comment to be created. The comment can extend over record boundaries, but the operator is not aware of this. It has rudimentary editing facilities.

Sessions

This unit contains the heart of the system. Its single entry sets up and runs a session. The unit will:

> Extract from the current game, slide, toy and utterance records all information needed to run a session. The slides may be checked by the operator during this phase
>
> Check the equipment, and ask the operator to check that the toys are correct. Ask her to get the subject ready
>
> Set the chronograph and run the session, creating a new session record
>
> Call the statistical package, which will update the current statistical record, or create a new one

The second phase controls the sequence of utterances and slide changes, and notes the responses. All events are timed to the nearest tenth of a second. The sequence will change from game to game. A typical game is MATCH1. In this game, the subject is required to match stimuli to responses. The following table will describe some of the game parameters, and give the values used for a given subject.

| Parameter | Unit | Range | Default | Subject Value |
|---|---|---|---|---|
| Game length | tenths | 0 .. 32000 | 600 | 600 |
| No response delay | tenths | 0 .. 32000 | 100 | 100 |
| Class count | | 1 .. 8 | | 2 |
| Class names | | | | Yellow Blue |
| Class change if correct | x / 10 | 0 .. 10 | 5 | 5 |
| Class change if wrong | x / 10 | 0 .. 10 | 0 | 0 |
| Extra noise | Response count | | 0 | 5 |

This game requires a set of utterances, which again will vary from subject to subject. For this game, seven are required, as shown below. The subject values are shown as textual representations, and therefore cannot describe the intonation and voicing used for a subject.

| Utterance Class | Subject Value |
|---|---|
| Affirmative | Terrific |
| Negative | No, Dave, that was . . . |
| Extra | Don't DO that! |
| Class 1 | Yellow |
| Class 2 | Blue |

In this game, the session unit cycles through the following sequence until the game parameter time is reached, or the operator terminates the game.

1 Calculate new difficulty and class depending upon the old values, the game parameters, and the current right/wrong counts

2 Pick a slide which matches these criteria as closely as possible

3 Issue the relevant describing utterance, and show the slide, noting the time

4 Wait for a response. If the waiting period exceeds the response wait time in the game parameters,

assume it is wrong. Note the time

5 If the answer if correct, issue the affirmative utterance. If it is wrong, issue the concatenation of the negative utterance and the correct describing utterance ('No, Dave, that was', 'Blue')

6 If the extra utterance is required, and the relevant game parameter is reached, issue the utterance, and zero the count

7 Store the event in the session record. The slide, response, times, correct and answered class are stored. 'No response' is distinguished from 'wrong'

The session unit then invokes the statistics package, and exits.

The Synthesizer Unit

This unit is an extreme example of the ratio of cosmetic to algorithmic code. The code to drive the synthesizer is two lines of Pascal, while the code to drive an utterance is ten lines long. The rest of the three files of text is cosmetic.

The synthesizer and its controller are capable of sixty three phonemes, in four voices at fifteen different pitch levels, and fifteen volume levels. The utterance record consists of an array of these, enough for about ten seconds of speech, a text line, and a number to key the utterance to the game in use. The driving code is very simple.

The cosmetic code is complex. It includes a help facility, a picture with examples of the available phonemes, 'thermometers' for the current pitch, volume, and voice levels, and a display of the current utterance. There are commands to add, delete, change, and create phonemes within an utterance, and commands to sing an utterance, or a single phoneme. The unit can file utterances away on disk and retrieve them.

Only a few utterances can appear on the screen at a time, so facilities for windowing are available.

The Statistics Package

The statistical unit has three entries, one of which is called each session. This entry compresses the raw session data. The second entry is called from the main loop, and does cross-session processing. The third prints a statistical record.

The package maintains one record (or a set of continued records) per session set. It is implemented as a variant structure, so that different compressions can be kept. The print

entry is placed in this unit so that as new statistical requirements come to light, only this unit need be changed. Currently, only one compression is available: a simple count of correctly, incorrectly and unanswered stimuli, and the average response time. New compressions are easy to add.

The cross-session entry is rudimentary. It simply prints a statistical record in a different form.

The Carousel and Timing Unit

This unit combines the projector driving, response detecting and timing functions. The key carousel facility is displaying a slide. One can choose to move and display, or just move. In the latter case, the program can drop the slide quickly, as it has already been moved to the correct place.

The responses are currently simple switches, and the controller is simply a buffer. For some input devices, such as joysticks, the 'jitter' information when the subject has poor fine motor control, can be used to interpret the subject's intent. For this reason, no debouncing is done in the controller. In the present case, debouncing is all the signal processing that is required, and it is done in software in this unit.

The timing facilities model a wrist chronometer; there is a timing register, and a clock register. The timing register can be stopped, started, and the current 'lap' time noted. The clock register, once set up, simply records the passing of real time. Both work to the nearest tenth of a second.

The Library and Date Units

It is imperative that the program performs its own error recovery. As an example, an incorrect keyboard input, such as an alphabetic when a numeric is called for, should never cause a system error. For this reason, a library unit was constructed with several handy functions. They include:

a bullet-proof number reader

a limit checking number reader

a length checking string reader

a 'prompt' routine

a 'yes-no' function

The industry has seen hundreds of variations on the syntax of a date, and we have implemented another. The date unit provides this function, as well as functions to read and set the

system date, functions to extract the day of the week from a date, and 'day number' functions so that the number of days between two dates can be calculated.

## 6 Conclusions and Suggestions for Further Work

The equipment has been delivered to Turner Village Hospital. The training of operators is complete, and took little time. Trial runs with residents at the hospital have been successful. Full clinical studies are in the process of being designed by the psychology staff. It is the results of these studies that will prove whether this work is truly successful, but the description of them is beyond the competence of this author. It is safest to say at this point, that the psychologists and care staff are happy with the system, and with its capabilities. The rest of this section will indicate how well the system met its design criteria, and suggest further work.

The stimuli, (that is the projected pictures and utterances) appear to be adequate for the teaching task required of them. The projector changes slides just fast enough in the worst case to keep the interest of the subject. The slides, of course, are limited to static images.

The voice synthesizer is surprisingly successful. One operator characterized the synthesizer output as 'a cross between Henry Kissinger and a Dalek'. Nevertheless, those subjects with some understanding of human speech have little trouble understanding and reacting to the sounds. Those who react to intonation only take a little longer to train.

It is clear that video output would be an improvement. A random access video disk would allow the recording of vignettes as well as stills. Recording would take little more skill than photography does now, or perhaps less, as there would be instant feedback. The utterances would be recorded speech using the same recording medium. This solution is well beyond the current budget.

The response devices have been reliable. Large 'toys' do cause some problems, as they can roll off the buttons. Fixing the buttons on boards would correct this, but make them less flexible to use. The current mechanism is only adequate for subjects with reasonable gross motor control.

The generic games installed appear more that adequate for the pilot study. Most new games suggested by the psychologists have been trivial to install. However, the implemented data structures do not cope with complex games. An example is a relationship game (eg under-over, bigger-smaller) in which each slide carries more than one relationship (a big ball over a small table). For this game, an utterance of the form "'Is the', 'ball', 'over', 'the table'" would have to be constructed. There

BEST COPY AVAILABLE


is scope here for the application of the techniques of expert systems.

All the information specified in the design criteria is kept, with the individual-to-individual flexibility specified. It provides the constant environment required. The system copes well with the problems of archiving and data integrity. Once the system is bootstrapped, it appears as a turnkey system. If mistakes are made, they do not destroy data. They may leave superseded information in the archive which has no value.

The statistical output is the weakest part of the system. A decision was made not to implement elaborate graphical output until the study had run for some time, and the psychologists were clear about the statistics needed. As the raw data is never lost, this is not a problem. The relatively primitive current implementation is easy to augment. It appears to be adequate for the pilot study. This is a very fruitful area for further work.

The test of the cosmetics has been the operator training time. It has averaged three hours. The one area where a change in 'style' may be required is in creating utterances. Perhaps a better solution would be to install an utterance in a 'Dalek'-like way, and then use a finger on a touch sensitive screen to vary the pitch and volume. This would be fruitful area of work for someone interested in Man-Machine Interfaces.

The programs have been well integrated into the turnkey system with the chainer unit. The structure seemed adequate during development; there is very little cross-coupling between units. As an example, an experiment was made with the installation of a blow tube instead of the switches. The carousel unit changes were not difficult, and no other changes were needed.

The system appears to be a success in its own terms. Subjects enjoy it, and even though at the time of writing, the full pilot study is not in progress, are gaining from it. It is easy to use, and reliable. Nevertheless, the test of its usefulness will be on the wards and in ordinary living. If improvements here can be attributed to the use of this system, then it will be truly a useful teaching tool.

Bibliography

Precision Teaching and Related Subjects

Altal: Cameron, D R et al 1982. Dependent Handicapped Curriculum Guide Edmonton Alberta: Alberta Education

Bate1: Bates, D F School-wide Implementation of Precision Teaching (Doctoral dissertation, University of Kansas 1971). Dissertation Abstracts International, 1972, 32, 548A

Bate2: Bates, S and Bates D F 1971. '. . . And a Child Shall Lead Them': Stephanie's Chart Story. Teaching Exceptional Children 3, pp 111-113

Hirs1: Hirsch, J 1971. Let Your Students Chart Your Behaviour. Teaching Exceptional Children 3, pp 135-136

Lind1: Lindsley, O R 1971. Theoretical Basis for Behaviour Modification. In Pitts, C E (ed) Operant Conditioning in the Classroom New York: Thomas Y Crowell Co, pp 54-80

Lind2: Lindsley, O R 1972. From Skinner to Precision Teaching: The Child Knows Best. In Jordan, J B and Robbins, L S (eds) Lets Try Doing Something Else Kind of Thing Arlington VA: Council for Exceptional Children, pp 1-11

Lind3: Lindsley, O R 1970. Children are not Retarded. In Fargo, G A et al (eds) Behaviour Modification in the Classroom Belmont CA: Wadsworth Publishing Co Inc, pp 310-16t

Meff1: Mefford, J P The Efficacy of a Precision Teaching Approach with Trainable Mentally Retarded Children (Doctoral Dissertation, Indiana University 1972) Dissertation Abstracts International 1973, 33, 6206A-6207A

Mont1: Wolking, W D et al 1981. Precision Teaching Project Training Manual Great Falls Montana: Great Falls Public Schools, 5

Perr1: Perry, B F 1974. The Liberation of Johnny. Teaching Exceptional Children 6, pp 154, 161

Pitt1: Pitts, C E (ed) 1971. Operant Conditioning in the Classroom New York: Thomas Y Crowell Co

Implementation Related Subjects

Ada1: Ada Joint Programme Office 1982. Reference Manual for the Ada Language Washington DC: US Department of Defence

Dihk1: Dijkstra, E W 1965. Solution of a Problem in Concurrent Programming Control, CACM, 8 pp 569

Elov1: Elovitz, H S et al 1976. Letter-to-Sound Rules for Automatic Translation of English Text to Phonetics. IEEE Transactions on Acoustics, Speech and Signal Processing, 6

Jens1: Jensen, K and Wirth, N 1978. Pascal Users Manual and Report, 2 New York: Springer-Verlag